

**ARQUITECTURA LIMS (LABORATORY INFORMATION MANAGEMENT
SYSTEM) PARA GESTIONAR EL USO DE RECURSOS Y ESPACIOS PERTINENTES
AL GRUPO DE INVESTIGACIÓN SIRIUS DE LA UNIVERSIDAD TECNOLÓGICA
DE PEREIRA**

LAURA SOPHIA GONZÁLEZ MONTOYA

JORGE LUIS SÁNCHEZ OCAMPO

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

PEREIRA

2020

**ARQUITECTURA LIMS (LABORATORY INFORMATION MANAGEMENT
SYSTEM) PARA GESTIONAR EL USO DE RECURSOS Y ESPACIOS PERTINENTES
AL GRUPO DE INVESTIGACIÓN SIRIUS DE LA UNIVERSIDAD TECNOLÓGICA
DE PEREIRA**

LAURA SOPHIA GONZÁLEZ MONTOYA

JORGE LUIS SÁNCHEZ OCAMPO

INFORME DE PROYECTO DE GRADO DE PREGRADO

DIRECTOR:

RAMIRO ANDRÉS BARRIOS VALENCIA

MAGISTER EN INGENIERÍA EN SISTEMAS

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

PEREIRA

2020

DEDICATORIA

“A mis padres y hermanos, quienes me han apoyado siempre en este camino, por darme la fuerza necesaria para seguir adelante.

Agradezco a mis docentes y director de proyecto de grado Ramiro Andrés Barrios, por ayudar en mi formación como profesional y como persona.”

JORGE LUIS SÁNCHEZ OCAMPO

“A mi madre Nora Elena y a mis abuelos Ofelia y Didier por su amor, apoyo y esfuerzo incondicional, que a pesar de las adversidades nunca me permitieron rendir, sin ustedes nada de esto sería posible.

*A mis compañeros por su cariño y apoyo constante durante este proceso.
A mi director de proyecto Ramiro Andrés Barrios por la confianza depositada en mí, junto a su guía y enseñanzas”*

LAURA SOPHIA GONZÁLEZ MONTOYA

INDICE

RESUMEN	7
TÍTULO	8
INTRODUCCIÓN	9
DEFINICIÓN DEL PROBLEMA	10
3.1 ANTECEDENTES	10
3.2 FORMULACIÓN DEL PROBLEMA	10
3.3 DESCRIPCIÓN DEL PROBLEMA	10
JUSTIFICACIÓN	11
OBJETIVOS	12
5.1 OBJETIVO GENERAL	12
5.2 OBJETIVOS ESPECÍFICOS	12
MARCO REFERENCIAL	13
6.1 MARCO CONCEPTUAL	13
6.2 MARCO DE ANTECEDENTES	14
6.3 MARCO LEGAL	17

	1
DISEÑO METODOLÓGICO	18
7.1 Requerimientos	18
7.2 Diseño e implementación	18
7.3 Documentación	19
7.4 Pruebas	19
CRONOGRAMA	20
ANÁLISIS DEL SISTEMA	23
9.1 DESCRIPCIÓN DEL PROYECTO	23
9.2 DESCRIPCIÓN DEL SISTEMA	23
LIMITACIONES	24
ALCANCE	24
DISEÑO DEL SISTEMA	25
12.1 Características del producto	25
12.2 USUARIOS	27
12.3 Fase de Requerimientos y análisis	28
12.3.2 Diagrama de casos de uso	32
12.4 FASE DE DISEÑO Y DOCUMENTACIÓN	40
12.4.2 Diagrama de clases	41
12.4.3 Tarjeta CRC	42
12.4.4 Diagrama de despliegue	43

	1
12.5 Fase de implementación	44
12.5.1 Lenguaje de programación Javascript	44
12.5.2 IDE de desarrollo	44
12.5.3 Motor de base de datos	44
12.5.4 Framework	45
12.5.5 Plantilla estándar de codificación	46
TRABAJOS FUTUROS	51
PRESUPUESTO	52
BIBLIOGRAFÍA	53

Lista de ilustraciones

Figure 1.DC.Inicio de sesión	33
Figure 2.DC.Gestión de Recursos	34
Figure 3.DC.Gestión de espacios	35
Figure 4.DC.Gestión de usuarios	36
Figure 5.DS.Inicio y Cambio Contraseña	38
Figure 6.DS.Solicitar Préstamos y Entregar Recurso	38
Figure 7.DS.Solicitar Préstamos y Entregar Espacio	39
Figure 8.DS.Actualizar datos del perfil	40
Figure 9.DBD.Diagrama de base de datos	41
Figure 10.Diagrama de clases	42
Figure 11.Diagrama de despliegue	44

Lista de tablas

Tabla 1.Cronograma	26
Tabla 2.Características del producto	30
Tabla 3.Perfiles Interesados	30
Tabla 4.Perfiles usuario	31
Tabla 5.Historia de usuarios	31
Tabla 6.Historias de Ususario 2	32
Tabla 7.Historias de Ususario 3	32
Tabla 8.Historias de Ususario 4	33
Tabla 9.Historias de Ususario 5	33
Tabla 10.Historias de Ususario 6	33
Tabla 11.Historias de Ususario 7	34
Tabla 12.Historias de Ususario 8	34
Tabla 13. CRC Gestionar Usuarios	45
Tabla 14. CRC Usuarios	45
Tabla 15. CRC Gestionar Recursos	46
Tabla 16. CRC Gestionar Espacios	46
Tabla 17. Reglas de estilo JavaScript	51
Tabla 18.Regla presentación JavaScript	53

RESUMEN

Este proyecto tiene como objetivo el desarrollo de un Sistema de Gestión de Información para el grupo de investigación SIRIUS de la Universidad Tecnológica de Pereira, con el fin de dar solución a la problemática sobre procesos críticos como préstamo o reserva de espacios y recursos, para esto se plantea, un software basado en la arquitectura LIMS para laboratorios cuyo propósito es el de gestionar y administrar de manera óptima los espacios y sus recursos dentro del grupo de investigación.

Por medio de la investigación cualitativa en una primera fase, se consultaron documentos que permitieron ver las implementaciones, que ya se han realizado acerca de los sistemas de información con la arquitectura LIMS, posterior a ello con la aplicación de entrevistas se logró registrar los procesos críticos a optimizar en la gestión de recursos y espacios.

Para la fase de Diseño e Implementación, el resultado obtenido fue el modelado del sistema, sujeto a las tecnologías de desarrollo utilizadas y los requerimientos de la fase anterior, los cuales representan los procesos, objetivos, reglas, actores, eventos y actividades, integrados para implementar un prototipo operacional de la aplicación.

Tras los resultados de la fase anterior se desarrollaron manuales de usuario, los cuales son la guía para el correcto uso del Sistema de Gestión. Finalmente se realizaron pruebas que permitieron mejorar la experiencia del usuario.

De esta manera se concluye que con la puesta en funcionamiento de esta aplicación se logra una mejor gestión de los recursos y espacios, optimizando tiempos y ofreciendo información actualizada y al instante de la disponibilidad de estos.

TÍTULO

Arquitectura LIMS (Laboratory Information Management System) para gestionar el uso de recursos y espacios pertenecientes al grupo de investigación SIRIUS de la universidad Tecnológica de Pereira

INTRODUCCIÓN

Los laboratorios en general han descubierto la necesidad de llevar un mejor control de sus actividades, y los elementos relevantes para el desarrollo de las mismas; identificando la necesidad más específicamente de saber en todo momento que pasa dentro del laboratorio, esto ayuda a mejorar los procesos diarios internos. Tener presente que hace el laboratorio y que necesita para el cumplimiento de sus funciones, ayuda a medir la eficiencia y la eficacia con la que funciona, permitiendo de esta manera una mejora constante.

Dentro de este documento se abordará el desarrollo de un sistema de gestión de la información que pretende administrar recursos y espacios perteneciente al grupo de investigación SIRIUS, con el objetivo de mejorar la calidad de sus actividades. Se podrá encontrar en este documento trabajos previos de otros laboratorios y la metodología que se escogió para desarrollar este sistema. Se contempló una fase de requerimientos, una de diseño y desarrollo y finalmente una de documentación y pruebas. Todo esto con el fin de que el sistema desarrollado cumpla con el concepto LIMS (Laboratory Information Management System), y de esta manera permita a los integrantes del grupo, la gestión adecuada de los recursos.

Describimos más a detalle de que se trata lims, mediante el diseño y arquitectura del prototipo que se desarrolló para el grupo SIRIUS, donde se puede observar el funcionamiento del sistema al describir las tecnologías implementadas, y mostrando los diagramas UML y de secuencia donde se puede apreciar de manera más técnica, la interacción entre todas las partes que conforman el Sistema de Gestión.

DEFINICIÓN DEL PROBLEMA

3.1 ANTECEDENTES

Anteriormente ha habido implementación de LIMS dentro de algunos laboratorios de la universidad reflejados por medio de proyectos de grado, ambos por parte de la carrera de Ingeniería en sistemas, para el laboratorio de Biotecnología y productos de naturales de la facultad de Química y otro para el laboratorio de aguas, con la finalidad de optimizar la obtención de información dentro los informes y su correcta administración de estos. [1] y [2]

Estas implementaciones tuvieron una gran aceptación, como lo fue para el caso del laboratorio de aguas y alimentos. Disminuyeron los tiempos empleados en la creación de cotizaciones, además de que la búsqueda y gestión de estos era mucho más ágil. La solución planteada por medio de una arquitectura LIMS permitió la trazabilidad y gestión de procesos, además de la posibilidad de adaptarse a las necesidades futuras del laboratorio.

3.2 FORMULACIÓN DEL PROBLEMA

Dentro del grupo de investigación SIRIUS el manejo de recursos y de espacios no está propiamente administrado de manera que permita el manejo y la eficiencia del laboratorio y sus actividades.

3.3 DESCRIPCIÓN DEL PROBLEMA

Dentro del grupo de investigación SIRIUS no existe un sistema que administre y gestione de manera óptima los recursos y espacios pertenecientes a este. Por ende, el préstamo y reserva de estos, dificultan y retrasan los trabajos y procesos investigativos, aumentando los tiempo de entrega, generando pérdida de información, problemas de disponibilidad, entre otros. Esto representa un obstáculo para el correcto funcionamiento y desarrollo de las actividades necesarias por parte de los integrantes del grupo de investigación.

JUSTIFICACIÓN

En la Universidad Tecnológica de Pereira los grupos de investigación son pieza fundamental para el desarrollo profesional y personal de la comunidad estudiantil, dentro de estos grupos suelen encontrarse laboratorios y/o otros recursos físicos como instrumentos y herramientas para el desarrollo de sus procesos y actividades. Estos recursos deben ser gestionados y administrados de forma que sean de uso eficiente y óptimo para las actividades realizadas dentro de este. Existen a su vez arquitecturas como LIMS (Laboratory Information Management System) que se encargan, entre otras cosas, de administrar estos recursos, facilitando el acceso a ellos, evitando pérdida de tiempo y ayudando al optimizar el flujo de trabajo.

Los sistemas actualmente previstos sobre LIMS permiten desarrollar distintas actividades de autogestión de un laboratorio como el préstamo de recursos y espacios así mismo el conocimiento de cuando estos recursos están siendo ocupados y el cuándo volverán a estar disponibles para la comunidad.

La idea de este proyecto es desarrollar un modelo de LIMS que brinde estas propiedades de gestión para el grupo de investigación Sirius, y permita la integración del desarrollo presente a dicho ambiente. Se espera que dicho componente permita la correcta gestión y administración,

para lograr un flujo de trabajo óptimo, y de esta manera, ayudar a la mejora del desempeño de las actividades realizadas por el mismo.

OBJETIVOS

5.1 OBJETIVO GENERAL

Implementar un Sistema de Gestión de Información de Laboratorio (LIMS) para la administración de los recursos y espacios dentro del grupo de investigación SIRIUS del programa de Ingeniería de Sistemas y Computación de la Universidad Tecnológica de Pereira.

5.2 OBJETIVOS ESPECÍFICOS

- Recolectar la información acerca de los recursos y espacios disponibles para el ingreso dentro del LIMS con la ayuda del grupo de investigación SIRIUS
- Registrar las actividades concernientes que involucren los recursos y espacios a administrar.

- Integrar los datos recolectados dentro del sistema de administración de información.
- Realizar una fase de pruebas funcionales y no funcionales de nuestro software.
- Analizar el resultado y realizar retroalimentación del proyecto.

MARCO REFERENCIAL

6.1 MARCO CONCEPTUAL

El tema del manejo y administración responsable de los recursos de un laboratorio, entiéndase dichos recursos como equipos físicos e infraestructura dentro de las universidades, supone un reto de disciplina y organización que corresponde a llevar una carga extra a los encargados. Un sistema integral de gestión propone ayudar a disminuir esa carga, y un LIMS permite automatizar estos procesos, mejorando efectivamente la eficiencia de un laboratorio, Además de proporcionar registro para auditorías y rastreabilidad y a mejorar el cumplimiento regulatorio que todo proceso o actividad debe llevar. Actualmente los sistemas LIMS están presentes en los laboratorios de muchas compañías por dos sencillas razones: economía y usabilidad [18]. La eficiencia de este dentro de los laboratorios como herramienta para el soporte de la información manejada dentro de ellos, garantiza la mejora en las actividades dispuestas y apuesta a incrementar las posibilidades de éxito de las tareas propuestas. Sin embargo, en el pasado, los sistemas LIMS no ofrecían más del 30-40 por ciento de las funcionalidades específicas

adaptadas a las necesidades del usuario, esto por el coste alto que involucra la creación y mantenimiento de un software genérico. Ahora, con el paso del tiempo este factor ha ido disminuyendo con la aparición de lenguajes de programación como el interpretado pero manteniendo un coste alto por mantenimiento entre otros. Los sistemas LIMS pueden adaptarse y aunque esto genere un coste en la industria la posibilidad de integrarse con otros sistemas como los ERP (Administrador de recursos empresariales) o los ELN (Cuadernos electrónicos de laboratorio) deja mucho a favor sobre estos sistemas que continúan en la industria creciendo día a día [19]

Este sistema dentro del grupo pretende ser una herramienta que ayude en labores de administración, como es el caso de préstamo tanto de recursos como de sus espacios de investigación, reservando los tiempos de uso y liberando para cuando no lo estén.

El desarrollo de este sistema será sobre el framework de MERN, este es un stack en el que se usa JavaScript tanto en el cliente como en el servidor es decir Full Stack JavaScript. MERN es una de las variaciones de la pila MEAN (MongoDB Express Angular Node) donde el marco de interfaz o frontend es Angularjs tradicional y se reemplaza por Reactjs. React permite crear interfaces complejas a través de componentes simples y conectarlos a datos desde un servicio de backend y representarlos como html, react es un punto fuerte para el manejo de interfaces y para este proyecto es punto clave en la obtención de información por parte del usuario. Por otro lado, en el siguiente nivel en el marco del servidor se encuentra Express.js que se ejecuta dentro de un servidor Node.js de código abierto, este marco es el motor del framework ya que él gestiona la

información desde la base de datos conteniendo la lógica del negocio y la transmitiendo todo esto hasta el nivel de interfaz para ser visualizada por el usuario, por último MongoDB siendo esta una base de datos no relacional que a diferencia de la relacional no tiene un identificador que sirva para relacionar dos o más conjuntos de datos, lo cual permite el manejo de datos como documentos, facilitando el registro de información y garantizando flexibilidad en cuanto a crecimiento del software permitiéndose ser robustas y menos vulnerable.

6.2 MARCO DE ANTECEDENTES

- En la Universidad Tecnológica de Pereira se implementó la arquitectura LIMS, hacia el año 2017 para gestionar la calidad en la gestión de información en el laboratorio de aguas y alimentos. Como estrategia para disminuir la ineficiencia en el procesamiento de los datos y aumentar la productividad de los laboratorios que hacen parte del sistema integral de gestión de calidad de la Universidad Tecnológica de Pereira [1]

- En el año 2018 en la Universidad Tecnológica de Pereira, se plantea un sistema de gestión de conocimiento que permite a cada uno de los laboratorios de investigación gestionar todo el material intelectual producido con el fin de ofrecer aspectos como la correctitud, completitud, persistencia, seguridad y mínima redundancia de la información.[2]

- El sistema integral de gestión estratégica de laboratorios (SigeLAB) ayuda a la Universidad del Valle al manejo y administración responsablemente de los recursos de sus laboratorios, entendiendo dichos recursos como la infraestructura física, el personal y los equipos de laboratorio. Este sistema nace inicialmente como la herramienta adecuada para hacer visible la información de ubicación y contacto de nuestros laboratorios, sin embargo, rápidamente se extendió en funcionalidades para suplir otras necesidades identificadas, que a futuro permitirán generar indicadores de gestión para ampliar el panorama y facilitar la toma de decisión en cada inversión.

Actualmente el sistema cuenta con un catálogo de servicios de laboratorio, el inventario interno de equipos y el sistema de solicitudes de servicio (tanto aquellas realizadas por el personal de la institución, como las realizadas por usuarios externos o no pertenecientes a ella). [5]

- Ip6 LIMS (Laboratory Information Management System) es una aplicación informática estándar configurable según los requerimientos del usuario que, automatiza la organización del trabajo para obtener la máxima eficiencia desde la recepción de las muestras, a la liberación de lotes, consultas, boletines, control de calibraciones, etc.

La solución de Gestión de laboratorios y ensayos Ip6 LIMS permite controlar la información en todas las áreas del laboratorio y está construida sobre una arquitectura escalable orientada a datos y totalmente parametrizable. Es válida, tanto para laboratorios

de pequeñas y medianas empresas con pocos usuarios, como para laboratorios de grandes empresas con cientos de usuarios.[6]

- El sistema de gestión de información de laboratorio (LIMS) de LabWare ofrece servicio al cliente, racionalizar las operaciones de laboratorio, adaptarse con agilidad a las condiciones cambiantes del negocio, y cumplir con los estándares normativos. Durante más de 25 años, LabWare ha adoptado la filosofía de colaboración con sus clientes, lo que ha resultado en una amplia gama de potentes funcionalidades para los laboratorios de diagnóstico clínico, biobancos, bio repositorios, laboratorios de salud pública, laboratorios veterinarios y una variedad de organizaciones que realizan investigación clínica como la industria farmacéutica, biofarmacéutica, organizaciones de investigación clínica (CRO), universidades e instituciones gubernamentales.

La solución clínica LIMS de LabWare combina las funcionalidades de un sistema de información de laboratorio tradicional (LIS) con capacidades integradas adicionales que satisfacen las necesidades de investigación clínica y biobanco.[7]

- MendeLIMS es una aplicación más de los sistemas LIMS en los laboratorios clínicos, específicamente en los proyectos de secuenciación de genoma. Para estudios genéticos.[8]

- openBIS (open Biology Information System) es otra solución que integra un libro de notas electrónico y un sistema LIMS, para la recolección y gestión de información

generada en los laboratorios biológicos. Permite a los investigadores documentar eficientemente su trabajo; describir materiales, métodos, recopilar datos crudos y analizados. El sistema viene con una interfaz web fácil de usar donde los datos se pueden agregar, editar, listar y buscar.[9]

De estos trabajos previos se analizó el propósito general que aborda cada uno, y la solución que aportaron. Esto permitió plantear los objetivos que debía cumplir el sistema desarrollado y comprender mucho mejor cómo interactúa el sistema con el ambiente, concluyendo que la arquitectura LIMS se adapta a los diferentes laboratorios de manera específica y cumpliendo la función requerida por cada uno. Después de este análisis fue más claro cómo se debía adaptar al proyecto para cumplir con la meta propuesta.

6.3 MARCO LEGAL

Estas son las normas del estado colombiano para el manejo de la información

- Ley 603 de 2000. Protección de derechos de autor.
- Ley estatutaria 1266 del 31 de diciembre de 2008. Disposiciones generales del Hábeas Data.

- Ley 1273 del 5 de enero de 2009. Creación del bien jurídico tutelado denominado “de la protección de la información y de los datos”.
- Ley 1341 del 30 de julio de 2009. Se definen conceptos de las tecnologías de la información.
- Ley estatutaria 1581 de 2012. Protección de datos personales.
- Decreto 1377 de 2013. Protección de datos, se reglamenta parcialmente la ley 1581
- ISO 9001:2015 especifica estándares para el manejo de los recursos dentro de un Sistema de Gestión de Calidad [4]

DISEÑO METODOLÓGICO

7.1 Requerimientos

En esta etapa se establecen los requerimientos, tanto funcionales como no funcionales a partir de debilidades o brechas dentro de la gestión, que servirán como base para

desarrollar el plan de trabajo. Ayudará a definir como el software dará solución al problema planteado y qué herramientas proporciona para esto mismo.

Con la toma de los requisitos se establecen restricciones y atributos de calidad con los que debe contar el sistema y características que garanticen la usabilidad, confiabilidad y eficiencia, que permitan demostrar la capacidad del software en mejorar los procesos y sus resultados. La planeación de un LIMS envuelve la definición de un flujo de información limpio, la estructura en la que se genera los datos y los requerimientos de usuario.

De esta manera con la información obtenida se define tanto el desarrollo del software y el cómo deberá de ser al momento de su ejecución

7.2 Diseño e implementación

En base a lo recolectado en la etapa anterior, esta etapa pretende la implementación de la arquitectura sobre la tecnología planteada. Se estructuran patrones de desarrollo y lenguajes de modelado.

La planeación y diseño proveen una oportunidad de anticiparse a los problemas y a educar a los usuarios sobre el proyecto. Imaginar las expectativas del sistema y el cómo será su flujo de trabajo mantendrá la línea del diseño y desarrollo acorde a las funcionalidades necesarias para lograr los objetivos del LIMS.

Como primera medida se realizará un diseño no funcional teniendo como objetivo el visualizar el correcto flujo de información, tras las pruebas y la validación correcta de este se implementará un diseño funcional basado en los puntos anteriores, solventando los factores críticos que pretende solucionar el software .

7.3 Documentación

En esta etapa el principal factor será la creación del manual de usuario que describe el funcionamiento entre los componentes del software que ayude al usuario a comprender el manejo y utilidades de este.

También se recolectan los demás diagramas (vistas, bases de datos, UML, etc ...) que se crearon para la documentación de los desarrolladores, de esta manera también se asegura el mantenimiento y mejora continua de la arquitectura.

7.4 Pruebas

En esta etapa se evaluará la calidad del software en cualquier momento del desarrollo de este. Es importante identificar las falencias a tiempo para mejorar la experiencia y usabilidad del usuario.

CRONOGRAMA

[illegible]

[illegible]

[illegible]

[illegible][illegible][illegible]

Para el cálculo del presupuesto de este proyecto, se hará en base a la hora de uso de un computador que está alrededor de los \$1000 pesos colombianos. El proyecto gastará 20 horas semanales entre los integrantes de este, esto implica un gasto de \$20000, veinte mil pesos semanales, para obtener un total de \$460000 cuatrocientos sesenta mil pesos por las 23 semanas que dura el proyecto. Estos gastos serán asumidos por los integrantes del proyecto.

ANÁLISIS DEL SISTEMA

Comentario [1]: En la metodología se indica la construcción y levantamiento de los requerimientos, sin embargo no se evidencia en el análisis del sistema.

En esta sección se plasma el cumplimiento de los requerimientos funcionales y no funcionales, por medio del diseño y la estructura planteada del sistema. Es apreciable además la relación de los tipos de usuario, sobre la aplicación.

9.1 DESCRIPCIÓN DEL PROYECTO

Comentario [2]: Se queda corto

Teniendo en cuenta los problemas anteriormente mencionados, este proyecto plantea la arquitectura de los módulos más críticos identificados, con el fin de impulsar la calidad de los procesos, y su correcta gestión mediante la arquitectura LIMS, en los laboratorios de HPC e ITS con los que cuenta el Grupo de Investigación SIRIUS. Mejorando los procesos de préstamo y reserva de espacios y recursos, brindando disponibilidad e

integridad de información detallada de los elementos del sistema. Esto es posible gracias al análisis que se hizo de proyectos LIMS realizados anteriormente y la toma de requerimientos.

9.2 DESCRIPCIÓN DEL SISTEMA

El sistema de gestión basado en la arquitectura planteada permitirá generar reservas sobre los recursos y espacios, contando cada uno con un módulo independiente de visualización, además de poder cancelar las mismas. También un manejo de usuarios administrador y básico para separar los privilegios dentro de la aplicación. Esto será posible mediante el uso del stack MERN anteriormente mencionado, con el cual se desarrollaran los siguientes módulos, accesibles desde panel izquierdo de la interfaz de la aplicación:

1. Usuarios: El sistema cuenta con una adecuada administración de los privilegios de usuario, que le permite al o los administradores controlar el acceso a la información y funciones del sistema, por parte de los demás usuarios básicos. Así como editar los datos de cada usuario, activar o desactivar sus cuentas a criterio.
2. Reservas: Este módulo permite mediante un calendario, muy cómodo de utilizar, realizar reservas de espacios y recursos individualmente, indicando los

horarios disponibles, y facilitando información específica, acerca de los horarios en estado de reserva o no disponibles.

3. Informes: El administrador podrá generar un informe con la fecha actual, que lista los elementos activos o inactivos, brindándole una herramienta que facilita la toma de decisiones, y llevar registro de lo que pasa en el laboratorio.

ALCANCE

El sistema abarca reservas de espacios y recursos, así como la gestión de usuarios y sus privilegios. También permitirá la cancelación de reservas y el de un reporte pdf sobre los recursos y espacios disponibles y no disponibles dentro del sistema

El prototipo resultante tendrá a cargo un número limitado de procesos a gestionar relacionados al uso de recursos y espacios con los que cuenta el grupo SIRIUS, aclarando también que el ingreso de los recursos y espacios al sistema se realizará por parte del grupo SIRIUS.

DISEÑO DEL SISTEMA

Comentario [3]: De acuerdo a lo mostrado en sesión de meet del funcionamiento de lo implementado hasta ahora, el diseño se corresponde. Me gustaría que en sesión de meet, pudiéramos revisar el cumplimiento de las observaciones hechas en dicha reunión anterior.

12.1 Características del producto

12.1.1 Objetivos

ON-1: Establecer niveles de acceso (administrador y usuario) para la administración del software.

ON-2: Administrar los recursos y espacios disponibles para los integrantes del grupo de investigación SIRIUS

ON-3: Realizar el préstamo de recursos y espacios

ON-4: Visualizar la información acerca de los recursos y espacios en estado de préstamo.

ON-5: Agilizar el proceso de préstamo de recursos y espacios para los integrantes del grupo de investigación Sirius

ON-6: Garantizar la seguridad de la información de cada usuario registrado en el sistema.

ID	DESCRIPCIÓN	PRIORIDAD	OBJETIVO
CAR-01	El sistema debe permitir realizar inicio de sesión para los usuarios	MEDIA	ON-1
CAR-02	El sistema debe permitir crear recursos y espacios	ALTA	ON-2 ON-5
CAR-03	El sistema debe permitir solicitar un recurso o espacio	ALTA	ON-3 ON-5
CAR-04	El sistema debe permitir ver la información de un recurso en préstamo (quien lo prestó y cuando es la fecha de entrega de este)	ALTA	ON-4 ON-5

CAR-05	El sistema debe permitir cerrar sesión a un usuario	MEDIA	ON-1
CAR-06	El sistema debe permitir administrar los usuarios, si y sólo si, el usuario es un administrador	MEDIA	ON-1 ON-6

Tabla 2. Características del producto

12.2 USUARIOS

12.2.1 Perfiles interesados

NOMBRE	DESCRIPCIÓN	RESPONSABILIDAD
Director grupo de investigación SIRIUS	Persona a mando de los procesos realizados dentro del grupo de investigación	Encargado de velar por alcanzar los objetivos propuestos
Universidad Tecnológica de Pereira	Institución donde está adscrito el grupo de investigación y sus recursos	Encargada de brindar los espacios y condiciones para el desempeño de los grupos de investigación
Investigadores dentro del grupo de investigación SIRIUS	Interesados en las labores investigativas que se llevan a cabo dentro del grupo de investigación SIRIUS	Personas encargadas de los procesos y desarrollos dentro del grupo de investigación

Tabla 3. Perfiles Interesados

12.2.1 Perfiles usuario

NOMBRE	DESCRIPCIÓN	RESPONSABILIDAD
Administrador	Persona encargada de procesos administrativos dentro del grupo de investigación	Encargado de gestionar la administración de los recursos, espacios y usuarios del software
Invitado	Persona dentro del grupo que cumple labores investigativas	Encargado de la gestión de espacios y recursos

Tabla 4.Perfiles usuario

12.3 Fase de Requerimientos y análisis

12.3.1 Historias de Usuario

Historia de Usuario
Número: 01
Tipo de Requisito: Funcional

Prioridad: Media
Objetivo de Negocio: 5
Descripción: El usuario debe poder acceder al software desde un navegador

Tabla 5.Historia de usuarios

Historia de Usuario
Número: 02
Tipo de Requisito: Funcional
Prioridad: Alta
Objetivo de Negocio: 3
Descripción: El usuario debe poder realizar préstamos de recursos y espacios dentro del software

Tabla 6.Historias de Usuario 2

Historia de Usuario
Número: 03
Tipo de Requisito: Funcional
Prioridad: Alta
Objetivo de Negocio: 1
Descripción: El usuario debe poder recuperar la contraseña de acceso al software

Tabla 7.Historias de Usuario 3

Historia de Usuario
Número: 04
Tipo de Requisito: Funcional
Prioridad: Media
Objetivo de Negocio: 5
Descripción: El usuario debe poder acceder a la información sobre un recurso o espacio en estado de préstamo

Tabla 8. Historias de Usuario 4

Historia de Usuario
Número: 05
Tipo de Requisito: Funcional
Prioridad: Media
Objetivo de Negocio: 1
Descripción: El usuario administrador puede gestionar los demás usuarios y dar privilegios o eliminar otros usuarios

Tabla 9. Historias de Usuario 5

Historia de Usuario
Número: 06
Tipo de Requisito: Funcional
Prioridad: Media
Objetivo de Negocio: 2,3
Descripción: El usuario debe poder devolver un préstamo realizado sobre un recurso o espacio

Tabla 10. Historias de Usuario 6

Historia de Usuario
Número: 07
Tipo de Requisito: Funcional
Prioridad: Media
Objetivo de Negocio: 2
Descripción: El usuario administrador debe poder crear recursos o espacios para su administración

Tabla 11. Historias de Usuario 7

Historia de Usuario
Número: 08

Tipo de Requisito: Funcional
Prioridad: Media
Objetivo de Negocio: 4
Descripción: El usuario debe poder generar un informe del estado de recursos o espacios en el momento

Tabla 12.Historias de Usuario 8

12.3.2 Diagrama de casos de uso

Inicio de sesión

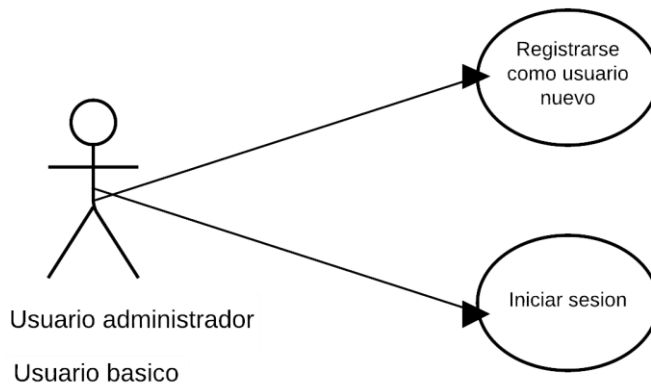


Figure 1.DC.Inicio de sesión

Gestión de Recursos

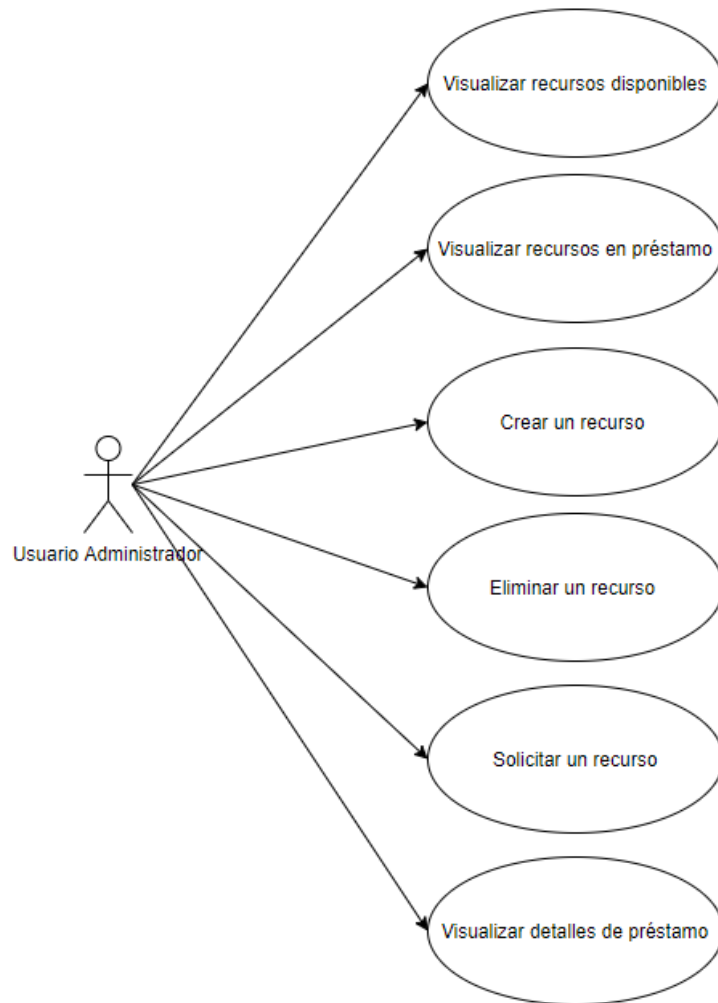


Figure 2.DC.Gestión de Recursos

Gestión de Espacios

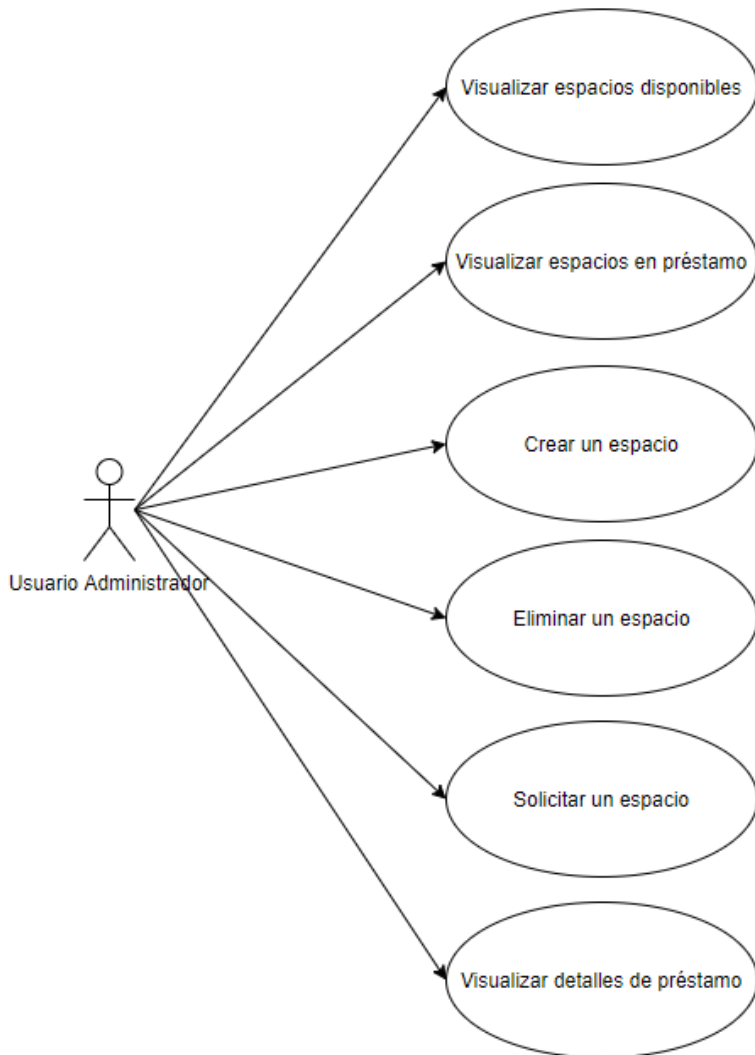


Figure 3.DC.Gestión de espacios

Gestión de usuarios

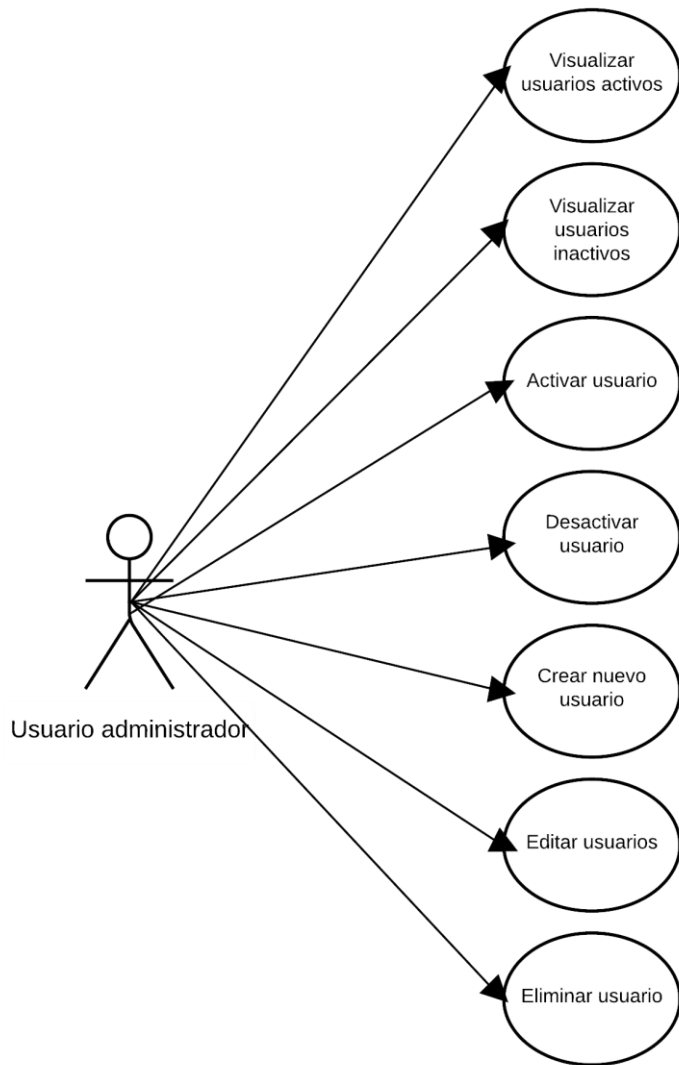
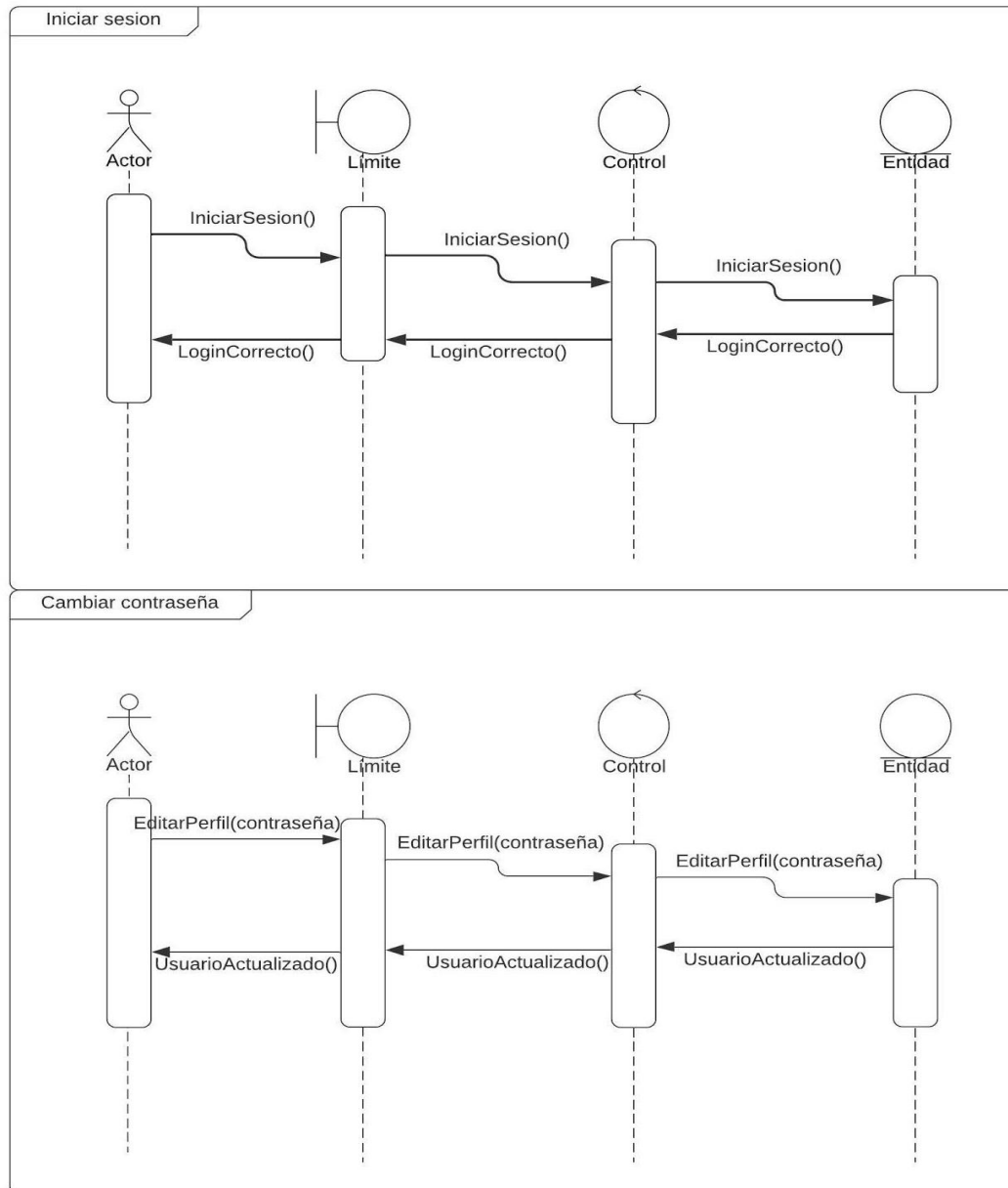
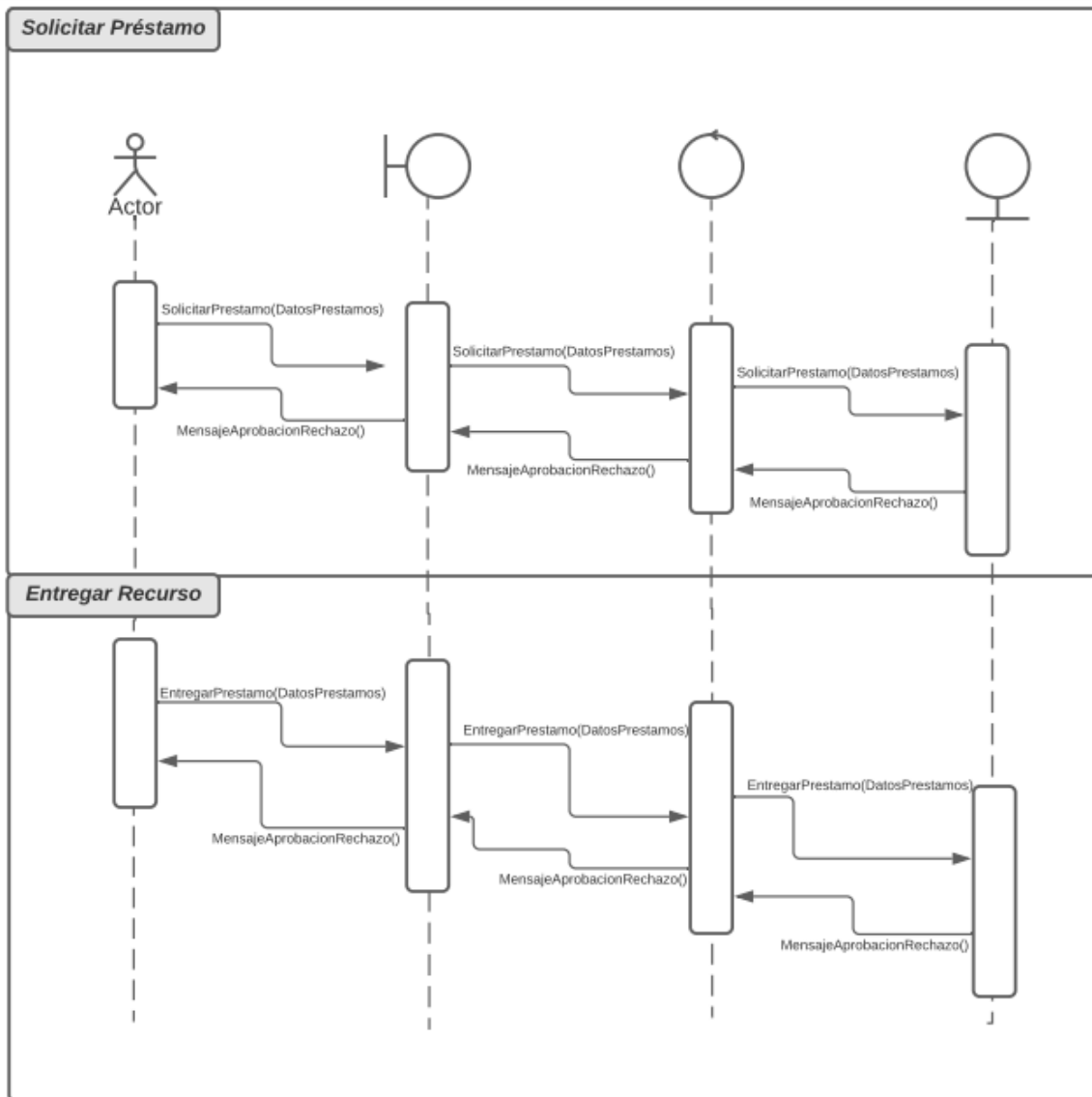


Figure 4.DC.Gestión de usuarios

12.3.3 Diagrama de secuencia

Iniciar sesión y cambiar contraseña



*Figure 5.DS.Inicio y CambioContraseña***Solicitar Préstamo y Entregar Recurso***Figure 6.DS.Solicitar Préstamos y Entregar Recurso*

Solicitar Préstamo y Entregar Espacio

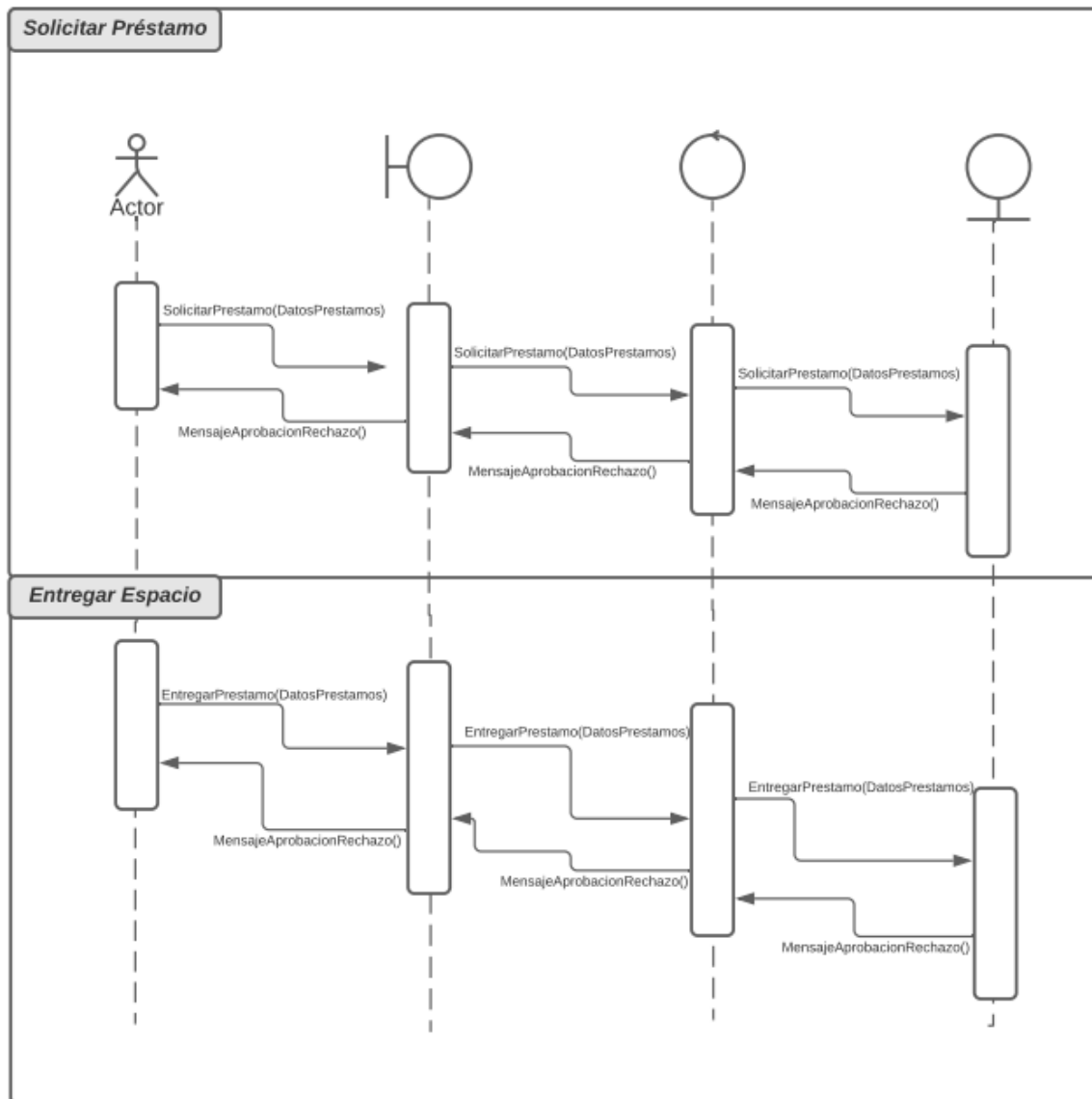


Figure 7.DS.Solicitar Préstamos y Entregar Espacio

Actualizar datos del perfil

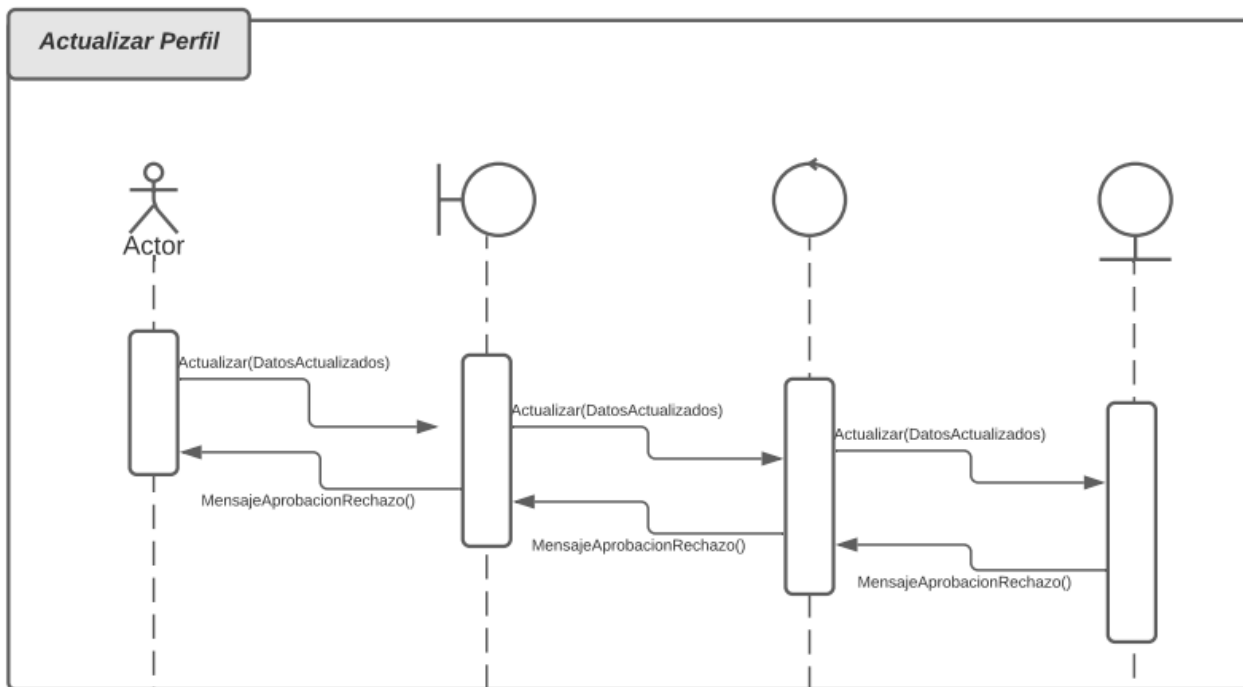


Figure 8.DS.Actualizar datos del perfil

12.4 FASE DE DISEÑO Y DOCUMENTACIÓN

12.4.1 Diagrama de base de datos

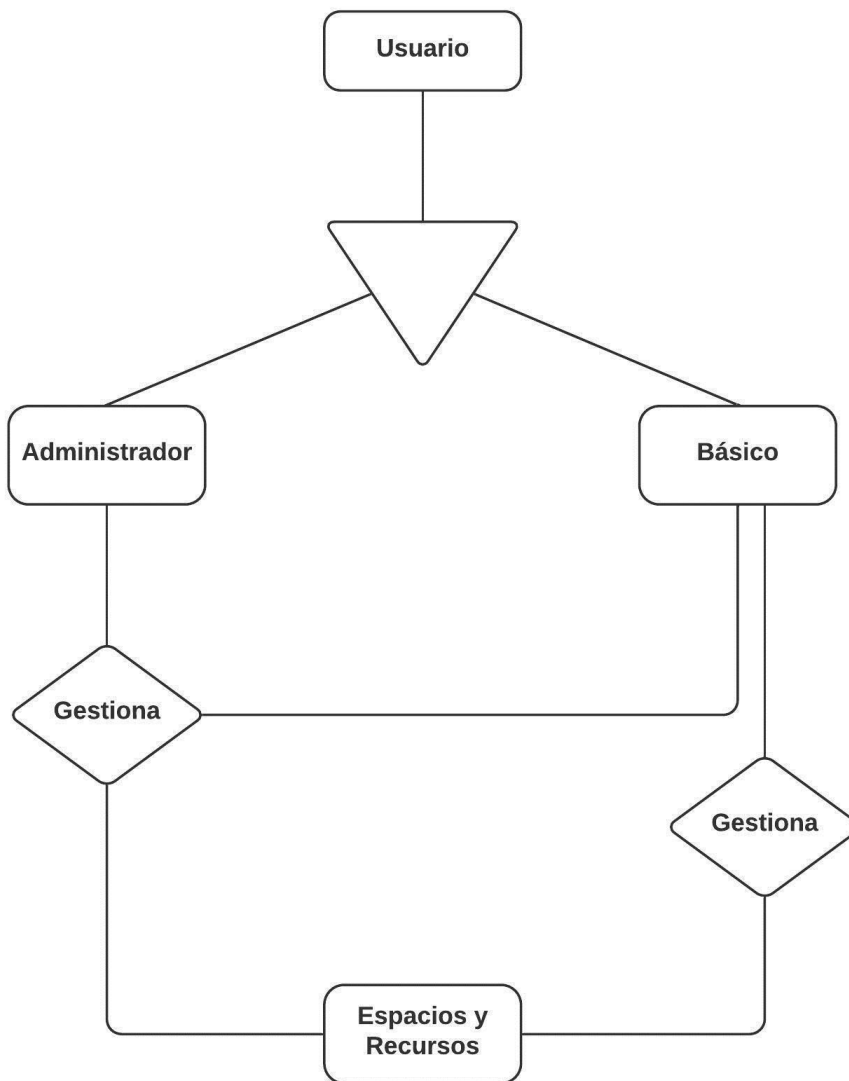


Figure 9.DB.Diagrama de base de datos

12.4.2 Diagrama de clases

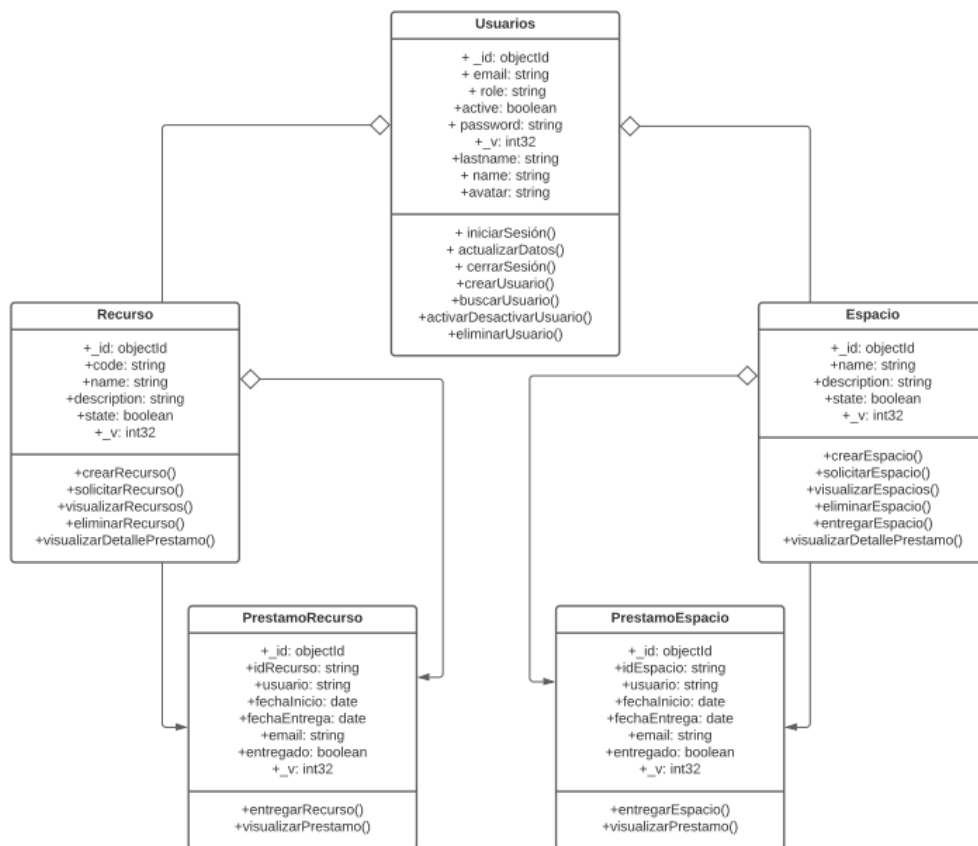


Figure 10. Diagrama de clases

12.4.3 Tarjeta CRC

Tarjeta CRC
<i>Nombre: Gestionar Usuarios</i>
<i>Responsabilidades:</i> <ul style="list-style-type: none"> ● <i>Gestionar la creación de usuarios</i> ● <i>Gestionar activación de usuarios</i> ● <i>Visualizar los usuarios activos e inactivos</i> ● <i>Gestionar edición y eliminación de usuarios</i>
<i>Colaboradores:</i> <ul style="list-style-type: none"> ● <i>Usuarios administradores</i>

Tabla 13. CRC Gestionar Usuarios

Tarjeta CRC
<i>Nombre: Usuarios</i>
<i>Responsabilidades:</i> <ul style="list-style-type: none"> ● <i>Solicitar apertura de una cuenta, para luego ser activada por el administrador del sistema</i> ● <i>Modificar información de perfil</i> ● <i>Iniciar sesión</i> ● <i>Cerrar sesión</i>
<i>Colaboradores:</i> <ul style="list-style-type: none"> ● <i>Todos los usuarios</i>

Tabla 14. CRC Usuarios

Tarjeta CRC
<i>Nombre: Gestionar Recursos</i>
<i>Responsabilidades:</i> <ul style="list-style-type: none"> ● <i>Gestionar la creación de recursos</i> ● <i>Gestionar el préstamo de recursos</i> ● <i>Visualizar los recursos según su estado</i>

<ul style="list-style-type: none"> • Gestionar entrega de recurso • Actualizar recursos en el sistema
Colaboradores: <ul style="list-style-type: none"> • Todos los usuarios

Tabla 15. CRC Gestionar Recursos

Tarjeta CRC
Nombre: Gestionar Espacios
Responsabilidades: <ul style="list-style-type: none"> • Gestionar la creación de espacios • Gestionar el préstamo de espacios • Visualizar los espacios según su estado • Gestionar Entrega de espacios • Actualizar espacios en el sistema
Colaboradores: <ul style="list-style-type: none"> • Todos los usuarios

Tabla 16. CRC Gestionar Espacios

12.4.4 Diagrama de despliegue

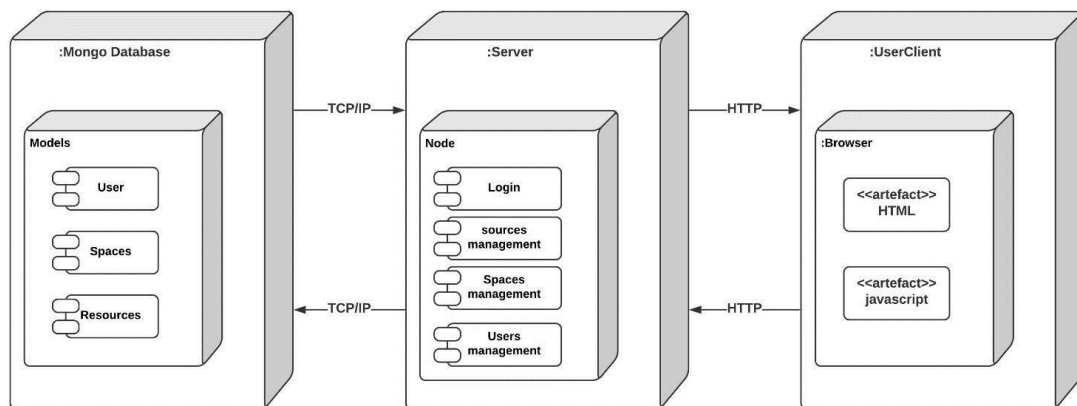


Figure 11. Diagrama de despliegue

12.5 Fase de implementación

12.5.1 Lenguaje de programación Javascript

Para la realización del software y por su naturaleza, Javascript es un lenguaje de programación utilizado en entornos dentro y fuera del navegador. El estándar para javascript es ECMAScript soportado en todos los navegadores modernos y algunas versiones para navegadores viejos. Además de servir tanto para el lado del cliente como del servidor[11][12].

12.5.2 IDE de desarrollo

Visual Studio Code combina la simplicidad de un editor de código con lo que los desarrolladores necesitan para su ciclo principal de edición, compilación y depuración. Proporciona soporte completo de edición, navegación y comprensión de código junto con depuración liviana, un modelo de extensibilidad enriquecido e integración liviana con herramientas existentes[13].Depurar código directamente desde el editor. Inicie o adjunte sus aplicaciones en ejecución y depure con puntos de interrupción, pilas de llamadas y una consola interactiva, Además de trabajar con git y otros proveedores de SCM.[14]

12.5.3 Motor de base de datos

Con MongoDB Atlas, sus clústeres de auto reparación se componen de instancias de bases de datos distribuidas geográficamente para garantizar que no haya un solo punto único de falla.

MongoDB Atlas facilita el control del acceso a su base de datos. Las instancias de su base de datos se implementan en una nube privada virtual (VPC) única, para garantizar el aislamiento de la red. Otras características de seguridad incluyen listas blancas de IP o peering VPC, autenticación siempre activa, cifrado en reposo y cifrado en tránsito, administración de acceso sofisticada basada en roles y más.

MongoDB Atlas automatiza el aprovisionamiento, la configuración y la implementación de la infraestructura para que sus equipos puedan obtener los recursos de base de datos que necesitan, cuando los necesitan. Los parches y las actualizaciones de versiones menores se aplican automáticamente. Y cuando necesite modificar su clúster, ya sea para escalar horizontalmente o realizar una actualización, MongoDB Atlas le permite hacerlo con unos pocos clics sin necesidad de una ventana de tiempo de inactividad.[15]

12.5.4 Framework

MERN es el acrónimo de MongoDB, ExpressJS, React JS y Node JS. El stack MERN es una combinación de las tecnologías anteriores, todas basadas en JavaScript, utilizadas para crear aplicaciones web avanzadas. Es un marco de desarrollo de código abierto de pila completa, es decir, proporciona componentes completos de desarrollo de front-end a back-end. Mientras que MongoDB, Express JS y Node JS son los componentes comunes (entre MEAN y MERN); AngularJS es reemplazado por React JS en el stack MERN.

- MongoDB almacena los datos en formato JSON binario que permite el intercambio rápido de datos entre cliente y servidor.
- Express JS es un marco modular y liviano de Node JS, que ayuda a crear aplicaciones web. Se puede asumir que Express JS, se ejecuta en la parte superior del Node JS, administrando el servidor y sus rutas.
- React JS es una biblioteca de JavaScript de código abierto que se utiliza para crear interfaces de usuario, generalmente para aplicaciones de una sola página.
- Node JS es un entorno de ejecución de JavaScript de código abierto, multiplataforma. Está diseñado para ejecutar el código JavaScript fuera del navegador, en el lado del servidor. Node JS accede al Node Package Manager (NPM), que aloja una gran cantidad de paquetes privados y públicos, y también se utiliza para publicar proyectos de Node JS.[16]

12.5.5 Plantilla estándar de codificación

Reglas de estilo para el lenguaje de JavaScript

REGLA DE ESTILO	EJEMPLO
<i>Declarar siempre las variables usando var. No hacerlo dificulta interpretar en qué ámbito se encuentra la variable y puede ocasionar que una variable esté en ámbito global indebidamente.</i>	<code>var nodosDiv;</code>
<i>Usa siempre puntos y coma para</i>	<code>var mivar1 = function() {</code>

<p><i>delimitar el final de una instrucción o línea.</i></p> <p><i>Cuando una función funciona como una expresión, debe delimitarse con punto y coma final. En cambio, en las declaraciones de funciones no se usa punto y coma final (ver ejemplo)</i></p>	<pre>return true; }; // Aquí incluir punto y coma final function miFuncion1() { return true; } // Aquí no incluimos punto y coma final</pre>
<p><i>No declarar funciones dentro de bloques (condicionales, bucles...). Si se requiere crear funciones dentro de bloques, usarlas como expresiones que se asignan a una variable.</i></p>	<pre>if (x) { function miFuncion1() {...} // ¡incorrecto! } if (x) { var mivar1 = function() {...}; // ¡correcto! }</pre>
<p><i>Hay que tener cuidado a la hora de crear closures porque pueden generar referencias circulares que impliquen desbordamiento de memoria.</i></p>	<p><i>No utilizar:</i></p> <pre>function miFuncion1(element, a, b) { element.onclick = function() { /* código que usa a y b, crea el closure que mantiene una referencia a element a pesar de no usarlo, mientras que element mantiene una referencia al closure, circularidad */ }; }</pre> <p><i>Para resolver esta situación se usa:</i></p> <pre>function miFuncion1(element, a, b) { element.onclick = miFun2(a, b); } function miFun2(a, b) { return function() { /* código que usa a y b */ }; }</pre>
<p><i>No usar with</i></p>	<p><i>Evitar el uso de esta sentencia</i></p>
<p><i>Limitar el uso de this</i></p>	<p><i>this puede tener distintos significados según el contexto. Usarlo sólo cuando sea</i></p>

	<i>realmente necesario.</i>
<i>No usar for-in para recorrer arrays.</i>	<i>Para recorrer arrays debe usarse un bucle for tradicional.</i>
<i>Crear arrays usando sintaxis de literal en lugar de new Array</i>	<i>Recomendado: var a = [x1, x2, x3]; No usar: var a = new Array(x1, x2, x3);</i>
<i>Crear objetos usando sintaxis de literal en lugar de new Object</i>	<i>Recomendado: var a = { }; No usar: var a = new Object ();</i>
<i>Crea el javascript en archivos js independientes en lugar de embeberlo en ficheros html</i>	<i>Esto hará reusable el código y hará más pesada la carga de los archivos html.</i>
<i>Usa === y !== en condicionales para realizar comparaciones preferentemente sobre == y !=</i>	<i>if (a === b) { ... }</i>
<i>Evita el uso de eval</i>	<i>Busca alternativas y evita usar la función eval</i>
<i>Incluye código previendo una llamada a una función en la que falten parámetros</i>	<i>function myFunction(x, y) { if (y === undefined) { y = 0; } } Esto se escribe más fácil y cómodo así: function myFunction(x, y) { y = y 0; }</i>
<i>Reduce el acceso al DOM.</i>	<i>Si tienes que acceder varias veces a un elemento del DOM, no uses el acceso tipo getElementById repetidas veces. Úsalo una sola vez y almacena el resultado en una variable. De esta manera mejorar el rendimiento.</i>

Tabla 17. Reglas de estilo JavaScript

Reglas de estilo recomendadas para la presentación de código JavaScript:

<i>REGLA DE ESTILO</i>	<i>EJEMPLO</i>
Usar la sintaxis camelCase para declarar nombres de variables y funciones (primera letra minúscula e intercalar mayúsculas).	nombreDeFuncionAsi, nombreDeVariableAsi, nombreDeMetodoAsi,
Crear los nombres de funciones y variables usando sólo 26 letras (a hasta z, sin la ñ), diez números (0 al 9) y el guión bajo _ . Evitar uso de la ñ o signos como \$, %, &, etc. Tampoco uses el guión medio.	var anyo = 2089;
Si una variable no va a cambiar de valor nunca se declara con todas sus letras en mayúsculas.	var NUMERODEAVOGADRO;
Los nombres de ficheros se escriben completamente en minúsculas. En el caso de ficheros html, se usa la extensión html en lugar de htm.	nombredeficherotodominusculas.js otroNombre.html
Incluye las llaves en la misma línea de aquello que abren (no pongas las llaves por ejemplo debajo de una condición de un if, sino al lado)	if (condicion) { // ... } else { // ... }
Usa siempre indentado para aquello que está dentro de un bloque. Por ejemplo, el código dentro de un bucle for estará desplazado hacia la derecha. Usa siempre el mismo indentado.	for (var i=0; i<10; i++) { //aquí codigo }
Las líneas no se deben alargar tanto como para no ser visibles. Se usa la referencia de	esto.aquello.alli.hacer = function(unArgumentoParaLaFuncionDeNombre1

80 caracteres. La línea no debe tener más de 80 caracteres. Si se excede esta longitud, la línea deberá dividirse.	, unArgumentoParaLaFuncionDeNombre2 , unArgumentoParaLaFuncionDeNombre3 , unArgumentoParaLaFuncionDeNombre4) { // código };
Usa comillas simples en lugar de comillas dobles	var nombre = 'carlos';
Separa las sentencias en distintas líneas. No escribas una detrás de otra separadas por ;	var x=1; var y=2; // No var x=1; var x=2;
Usa espacios separadores: esto hace más fácil de leer el código	while(x<3&&y<5&&z==10){llamarA(); } //No while (x < 3 && y < 5 && z == 10) { llamarA(); } //Sí
El código debe estar correctamente comentado. Los bloques /* ... */ se usarán para documentación formal (explicación de funciones, parámetros que intervienen, etc. y los comentarios en línea // para aclaraciones puntuales.	Para documentar proyectos extensos se recomienda seguir unas normas de comentarios y usar herramientas específicas que crean la documentación de forma automática, como JSDoc.
Respetar el estilo del código en edición	Si nos encargan realizar unos cambios en un código que tiene un estilo, respetar el estilo de dicho código, aunque no coincida con el que nosotros usemos habitualmente. Es importante que dentro de un código se mantenga siempre el mismo criterio.

Tabla 18.Regla presentación JavaScript

Anexos

- Anexo 1. Manual de Usuario

TRABAJOS FUTUROS

Como trabajo futuro dentro del desarrollo y ampliación del software, se pretende incorporar la implementación de la aplicación tipo webhook como backend, que responde a robot de telegram desarrollada en un trabajo de grado anterior, y a partir del mismo objetivo de soportar la gestión de recursos de un laboratorio.

Otra propuesta de mejora e innovación, es la realización de gestión de recursos para proyectos específicos dentro del grupo de investigación, y de esta manera, tener módulos de préstamo donde los recursos están asignados por el tiempo de duración de dichos proyectos y en los tiempos que los integrantes los necesiten para su realización, no dejando de lado el hecho de que otros miembros del grupo de investigación, fuera del

proyecto puedan utilizarlos cuando estén disponibles, es parte de la propuesta de mejora dentro del software.

A partir de la propuesta anterior y teniendo en cuenta, que los recursos y espacios utilizados en los proyectos quedarán registrados, estos registros servirán como información para realizar presupuestos de futuros proyectos que se deseen realizar tomando como muestra dicha información, dando una perspectiva más sólida para la toma de decisiones.

Conclusiones

- El presente trabajo provee una solución LIMS (Laboratory Information Management System) sobre la gestión y administración de recursos y espacios, permitiendo mejorar y optimizar la organización y desarrollo de las actividades concernientes al grupo de investigación SIRIUS.
- Con la aplicación de este software se obtuvo una mejora en los tiempos de entrega, ya que se puede ver la disponibilidad de los espacios y recursos, proveyendo una rastreabilidad sobre los componentes de los laboratorios.
- Se logró recolectar información de los recursos, espacios y procesos clave del grupo SIRIUS, comprendiendo las necesidades de los laboratorios para aplicar una solución tecnológica.

- El framework MERN permitió un desarrollo hábil y fluido, además de lograr un código simple y limpio para futuras mejoras o mantenimiento del software.

BIBLIOGRAFÍA

Comentario [4]: La calidad de la bibliografía debería mejorarse. Existen muchas referencias a páginas y blogs de poco renombre.

[1] Repositorios de Proyecto de Grado UTP. *Arquitectura LIMS para gestionar la calidad en los laboratorios del sistema integral de gestión de calidad de la Universidad Tecnológica de Pereira*. Recuperado de: <http://repositorio.utp.edu.co/dspace/handle/11059/8262> Consultado: 22/08/2019

[2] Repositorios de Proyecto de Grado UTP. *Desarrollo del módulo de gestión del conocimiento bajo la arquitectura LIMS (laboratory information management system) para los laboratorios de investigación de la Universidad Tecnológica de Pereira*. Recuperado de: <http://repositorio.utp.edu.co/dspace/handle/11059/9182> Consultado: 22/08/2019

- [3] Grupo de investigación SIRIUS UTP. Acerca de. Recuperado de:
<https://sirius.utp.edu.co/about> Consultado: 22/08/2019
- [4] *Estándar para un Sistemas de Gestión de Calidad*, Recuperado de:
<https://www.isotools.org/normas/calidad/iso-9001>. Consultado: 21/10/2019
- [5] SIGELAB. El sistema integral de gestión estratégica de laboratorios de la Universidad del Valle . ref: <https://sigelab.univalle.edu.co/inicio>, Consultado 21/10/2019
- [6] ip6 LIMS.solución de Gestión de laboratorios y ensayos Recuperado de:
<https://www.labware.es/1gestion-de-laboratorios-y-control-de-calidad/> Consultado: 21/10/2019
- [7] “LabWare”.Sistema de gestión de información de laboratorio (LIMS) Recuperado de:
<https://www.labware.com/es/p/0217>, Consultado 21/10/2019
- [8] GRIMES, S y JI, H. MendeLIMS: a web-based laboratory information management system for clinical genome sequencing. En: BMC Bioinformatics. 27 Agosto 2014. DOI: 10.1186/1471-2105-15-290. ref: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4155081/> Consultado 21/10/2019
- [9] BARILLARI, Caterina, et al. Journal: openBIS ELN-LIMS: An open-source database for academic laboratories. En: Bioinformatics. 2015. vol. 32. ISSN1460- 2059.

[10] HalX: an open-source LIMS (Laboratory Information Management System) for small- to large-scale laboratories Received 22 July 2004 Accepted 13 January 2005

ref: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.412.8399&rep=rep1&type=pdf>

Consultado 2/08/2020

[11] Javascript ref: <https://developer.mozilla.org/es/docs/Web/JavaScript> Consultado 7/09/2020

[12] ¿Qué es Javascript? ref: https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Qu%C3%A9_es_JavaScript Consultado 7/09/2020

[13] Código de Visual Studio - Código abierto ref: <https://github.com/microsoft/vscode> Consultado 7/09/2020

[14] Código de Visual Studio - Código abierto ref: https://code.visualstudio.com/?wt.mc_id=DX_841432 Consultado 7/09/2020

[15] Atlas de MongoDB ref: https://www.mongodb.com/cloud/atlas/lp/try2?utm_source=google&utm_campaign=gs_americanas_colombia_search_brand_atlas_desktop&utm_term=mongo%20atlas&utm_medium=cpc_paid

[search&utm_ad=e&utm_ad_campaign_id=2030069987&gclid=CjwKCAjwNf6BRAwEiwAkt6UQi-j6zeAd2s5Be1CnBhJ3HxgskHeYH1jT8XdDFi8AVNKikKqCOB9HhoC-J4QAvD_BwE](https://www.google.com/search&utm_ad=e&utm_ad_campaign_id=2030069987&gclid=CjwKCAjwNf6BRAwEiwAkt6UQi-j6zeAd2s5Be1CnBhJ3HxgskHeYH1jT8XdDFi8AVNKikKqCOB9HhoC-J4QAvD_BwE)

Consultado 7/09/2020

[16] ¿Qué es MERN Stack? ref: https://medium.com/@blockchain_simplified/what-is-mern-stack-9c867dbad302

Consultado 7/09/2020

[17] JavaScript. Manual de estilo: reglas, convenciones. Cómo crear objetos, arrays.

Ejemplos, ejercicio (CU01191E)

ref:

https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=880:javascript-manual-de-estilo-reglas-convenciones-como-crear-objetos-arrays-ejemplos-ejercicio-cu01191e&catid=78&Itemid=206

Consultado 7/09/2020

[18] La importancia de los sistemas LIMS en los laboratorios

ref: <https://zemsaniaglobalgroup.com/la-importancia-de-los-sistemas-lims-en-los-laboratorios/>

Consultado 19/09/2020

[19] Gestión de datos sencilla en toda la empresa: Ventajas de las soluciones LIMS integradas y específicas ref: <http://apps.thermoscientific.com/media/SID/Informatics/PDF/Article-Integrated-and-Purpose-built.pdf>

Consultado 19/09/2020

[20] Bases de datos relacionales vs no relacionales ref: <https://www.pragma.com.co/academia/lecciones/bases-de-datos-relacionales-vs.-no-relacionales>

Consultado 19/09/2020

[21] Pila MERN Stack ref: <https://www.mongodb.com/mern-stack> Consultado 19/09/2020

[22] Product Review: Implementing LIMS: A “how-to” guide ref: <https://pubs.acs.org/doi/pdf/10.1021/ac0027082> Consultado 20/09/2020

[23] Considerations in selecting a laboratory information management system (LIMS) ref: https://atlab.com/wp-content/uploads/2018/12/Considerations_in_Selecting_LIMS.pdf

Consultado 20/09/2020

